Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 7, No. 1, 2023

Research on Fault Diagnosis Method of Rotary Machinery Based on Improved Transformer

Haijie Zhi[1, *], Jinkui Wang[1, a], Haitao Zhang[1, 2, a], Yongkang Hou[1, a], Qishun Yang[1, a]

[1] College of Engineering, China University of Petroleum (Beijing) Karamay Campus, Karamay, 834000, China
[2] College of software, Taiyuan University of Technology, Taiyuan, 030024, China
[a]The contributions of these authors to this paper are consistent
[*] Corresponding author: Haijie Zhi (Email: 1810365936@qq.com)

Abstract: In modern industry, rotating machinery plays a crucial role. These rotating machines are not only fundamental components of power generation and propulsion systems but also key factors for their efficient operation. Harsh operating environments often lead to failures in critical components like gears and bearings in rotating machinery, which can directly result in equipment malfunction. Therefore, authentic fault diagnosis of these primary building blocks in rotating machinery is of pivotal value for improving its reliability and safety during operation. This research proposes a fault diagnosis strategy for rotating machinery rooted on time series Transformer and validates the effectiveness of the proposed approach. Firstly, a time series Transformer model is designed, which incorporates modules like time series embedding, attention mechanism, and multi-layer perceptron to promptly approach 1D oscillatory motion signal data. Subsequently, the hypothetical model is trained and tested on multiple public datasets, and its fault diagnosis results are compared with existing achievements in the literature. The proficiency of the model is thoroughly verified, and the fault diagnosis performance of this proposed approach surpasses many existing fault diagnosis methods.

Keywords: Rotating machinery, fault diagnosis method, Transformer, attention mechanism.

1. Introduction

As the advancement of industry and technological developments, progressive industrial machinery has become increasingly functional and complex. Among them, rotating machinery is one of the most crucial components in various industrial fields such as aviation, aerospace, and maritime industries[1]. However, due to harsh operating conditions such as heavy loads, high temperatures, and high pressures, some critical elements of rotating equipment, Some examples are bearings and gears, are prone to failures during operation. These failures can have a notable effect on the normal operation of the equipment, giving rise to a decline in service performance and, in extreme cases, mechanical failure or disintegration. This poses serious risks to manufacturing quality and operational safety. Therefore, fault diagnosis and health watchfulness on rotating machinery are essential aspects in industrial system design and maintenance[2].

The theory and methods related to machine learning have undergone many years of development and have been extensively adopted in various fields, including fault diagnosis. Currently, prevailing machine learning applications used in the field of fault diagnosis include SVM, kNN, SOM, and Back Propagation Neural Network (BPNN) consisting of fully connected layers[3]. Yan et al. utilized VMD to manipulate vibration signals from bearings, combining time-domain and frequency-domain features to extract multi-attribute, time-varying spectrum domain properties. Subsequently, they employed the Laplacian Score (LS) to select and reduce the dimensionality of the extracted high-dimensional time-frequency domain features. The reduced features were used as inputs for Support Vector Machine (SVM), and the SVM criteria were refined using PSO. This approach achieved effective fault diagnosis of bearings and demonstrated favorable fault diagnostic performance. Janssens et al. proposed a bearing fault diagnosis tactics that combines the FCM with K-Nearest Neighbor (KNN). In this method, PSO (Particle Swarm Optimization) was also utilized to optimize the KNN algorithm, reducing computational complexity[4]. Comparative test findings demonstrated that this approach can properly classify the major fault groupings of bearings using a limited amount of fault data. Zhang et al. utilized the chaotic adaptive gravitational search algorithm and PSO to optimize the training process of the BPNN. They operationalized this approach to the fault diagnosis problem of motor drive systems and found that introducing adaptive gravitational constant decay and chaotic mapping during the training process can enhance the pattern recognition performance of BPNN[5].

This paper presents an improved temporal dataset Transformer-based fault diagnosis method for rotating machinery. Firstly, a temporal dataset tokenizer is designed to directly process 1D format vibration signal data, and a multi-head self-attention mechanism is utilized to establish the time series Transformer model. Subsequently, the model is trained and tested on publicly reachable experimental datasets, and the testing results across multiple datasets are provided to analyze and test the performance of the suggested methodology. Through this validation, fault diagnosis performance of this approach outperforms many existing fault diagnosis methods.

2. Time Series Transformer Model Based on Mechanical Fault Diagnosis

In experiments, the collected raw vibration signals are typically 1D format time series data. However, the native Transformer models widely used in NLP (Natural Language Processing) domain take tokenized text sequences as input, and the Vision Transformer (ViT) used in the computer vision domain processes 2D format RGB images with three channels. These Transformer models cannot directly handle 1D format

vibration signal data. within this chapter's scope, we propose a Transformer model unambiguously constructed to process raw vibration signal data, called Time Series Transformer (TST). As shown in Figure 1, the TST model comprises components such as time series tokenization, Transformer layers, and classifier layers[6].

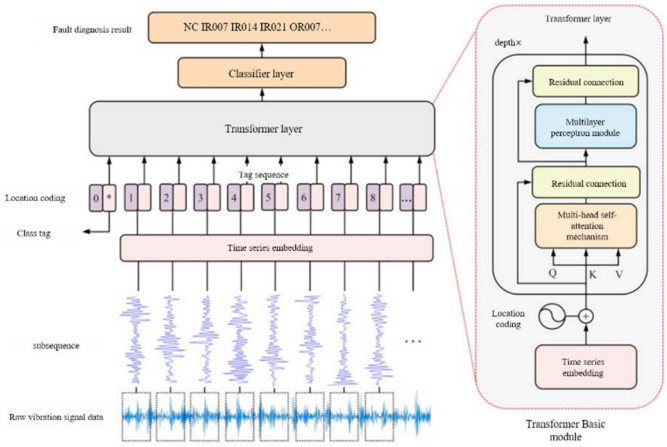

Figure 1. The composition of the planed TST paragon

2.1. Time series segmentation

2.1.1. Time series embedding

In general, a batch of time-ordered sequence can be represented as $t \in R^{B\times L}$, where B is the batch capacity, and L represents the examined time series' sampling length. In the process of time series transformation, initially, related to the tokenization process in NLP, the received time-ordered sequence is divided into sub-sequences of specified lengths. These sub-sequences are then concatenated in order to form a three-dimensional tensor, denoted as $[t^1_s, t^2_s, t^3_s, \ldots, t^{N_s}_s] \in R^{B\times N_s\times(L/N_s)}$, where N_s represents the number of segmented sub-sequences, and L/N_s should be an integer. Next, akin to the word embedding process, each sub-sequence is mapped to a high-dimensional transformation space through a simple rectilinear transformation, as shown in Equation (1).

$$TokensSeq = [t^1_s, t^2_s, t^3_s, \ldots, t^{N_s}_s] \bullet W_{embedding} \in R^{B\times N_s\times \dim} \quad (1)$$

Where $W_{embedding} \in R$ represents the linear mapping matrix, which is a attainable parameter, and *dim* denotes the embedding time series measurement. It should be noted that, in order to enable the proposed TST (Transformer-based Time Series) model to learn more general mappings and, concurrent with, reduce the number of model conditions, the TST model employs unvarying linear transformation matrix for all sub-sequences to perform the linear transformation.

2.1.2. Class tag

After feature extraction of marker sequences obtained by time series embedding based on attention mechanism, the TST model also needs to represent the extracted features and convert them into feature maps. At present, there are two main methods to represent the features of the tag sequence: (1) Global pooling of the entire tag sequence after the final Transformer layer. (2) Taking inspiration from BERT's processing approach, manually introduce category tags in the token sequence to achieve the conversion of the token sequence into feature maps. In this paper, the proposed TST model adopts the second method to derive the feature embedding. The class tag is actually a erratically arranged tunable parameter signified as $x_0 \in R^{1\times \dim}$. The sequence of tags after the class tag is added is shown in Formula (2).

$$TokensSeq = [x_0; [t^1_s, t^2_s, t^3_s, \ldots, t^{N_s}_s] \bullet W_{embedding}] \in R^{B\times(N_s+1)\times \dim} \quad (2)$$

2.1.3. Location coding

The proposed TST (Transformer-based Time Series) model lacks filtering-based operations and does not possess translational invariance derivation bias. Furthermore, the multi-head, self-attention mechanism does not include positional information of the entries during computation. As a result, the TST model neglects the positional relationships among various subsequences in the original vibration signal data during the computation process. To address this issue, it is necessary to preserve the absolute and relative positional information in the token sequence by incorporating the method of position encoding, as shown in Equation (3).

$$TokensSeq = [x_0; [t^1_s, t^2_s, t^3_s, \ldots, t^{N_s}_s] \bullet W_{embedding}] + E_{pos} \in R^{B\times(N_s+1)\times \dim} \quad (3)$$

Where E_{pos} indicates the location encoding.

2.2. Transformer layer

2.2.1. Multi-head self-attention mechanism

The Multi-head self-attention mechanism (MSA) is the most crucial component of the Transformer model and is entirely based on the attention mechanism, without involving any CNN or RNN-related structures. In general, the attention mechanism can be seen as a mapping from a query value (denoted as Q) to a set of key-value pairs (denoted as K and V, respectively), where Q, K, and V are all vectors. The input to the attention mechanism is a set of Q, K, and V values, and the output is the weighted sum of V, with corresponding weights represented by the Attention Distribution (AD) matrix. The essence of the AD is to represent the similarity between the two sets of sequences, Q and K. There are various ways to compute the AD, known as scoring functions. Currently used scoring functions include the Scaled Dot-Product Attention, Bahdanau Attention, and Cosine Similarity Attention, among

others. Among these, the Scaled Dot-Product Attention is widely used due to its simplicity in computation, ease of parallelization, and no introduction of additional parameters into the model. The specific computation of the Scaled Dot-Product Attention is as follows.

$$Attention(Q,K,V) = soft\max(\frac{QK^T}{\sqrt{d_k}})V \quad (4)$$

Where d_k indicates the entered dimensions of Q and K.

2.2.2. Multilayer perceptron module

To enable the proposed TST (Transformer-based Time Series) model to achieve more complex non-linear mappings, each Transformer module also includes a Multilayer perceptron block (MLP). The MLP module within the *l-th* Transformer module consists of a non-linear mapping layer with an activation function and a linear mapping layer, as shown in Equation (5).

$$MSP(y_l^{MSA}) = activation(y_l^{MSA}W_1^l + b_1^l)W_2^l + b_2^l \quad (5)$$

2.2.3. Classifier layer

The classifier layer is the final layer of the TST (Transformer-based Time Series) model in this paper, and it is responsible for converting the feature maps extracted by the TST model into one-hot encoded representations suitable for direct fault diagnosis. Its basic structure is similar to that of the MLP module, as shown in Equation (6).

$$ClassLayer(y) = y^{class} = Soft\max(yW^{class} + b^{classs}) \quad (6)$$

3. Data sets and Data Preprocessing Methods

In this paper, we utilized publicly available experimental datasets from the field of rotating machinery fault diagnosis to train, test, and validate the proposed TST (Transformer-based Time Series) model. The experimental datasets used include the Case Western Reserve University (CWRU) rolling bearing fault dataset, the Xi'an Jiaotong University (XJTU) rolling bearing fatigue fault dataset, and the University of Connecticut (UCONN) gearbox fault dataset. These datasets consist of three typical forms of rotating machinery faults: pre-fault rolling bearing, rolling bearing fatigue fault, and gearbox fault. The experimental equipment used for data collection for each dataset is shown in Figure 2. Below is an introductory glimpse into the datasets and the data conditioning methods used.

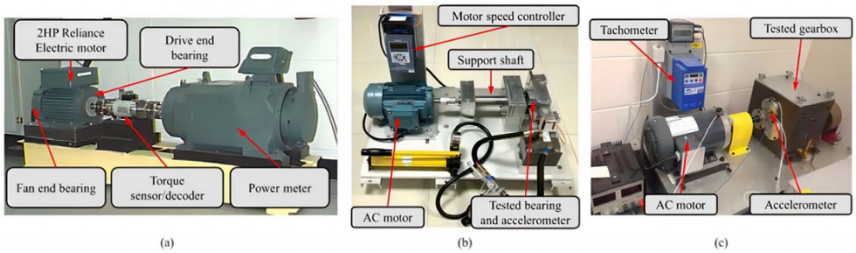

Figure 2. Schematic diagram of experimental equipment during data collection of each experimental data set (a) CWRU data set (b) XJTU data set (c) UCOON data set

3.1. CWRU data set

The CWRU dataset was collected with a high sampling frequency of 1.2kHz and 4.8kHz during the experimental process[7]. As a result, the original vibration signal data for each faulty bearing contains tens of thousands of data points. Due to their excessive length, these data cannot be directly used for training and testing the model. Therefore, a resampling method is employed to preprocess the data into a specified length of 2048 data points. The resampling method used in this chapter is illustrated in Figure 3 and mainly involves two sampling approaches. The first approach is to resample directly without any overlap in the original vibration signal data, which is the most common preprocessing method. The second approach is a special resampling method where overlapping regions are deliberately set during the sampling process, providing a form of data augmentation. Based on the original data from the CWRU dataset and the resampling approaches depicted in Figures 2-3, we constructed a CWRU dataset containing 9000 samples. Among these, 7000 samples form the training set, used to train the TST model proposed in this chapter. The remaining 2000 samples compose the test set, utilized to evaluate the fault diagnosis performance of the TST model. There is no overlap between the training and test sets.

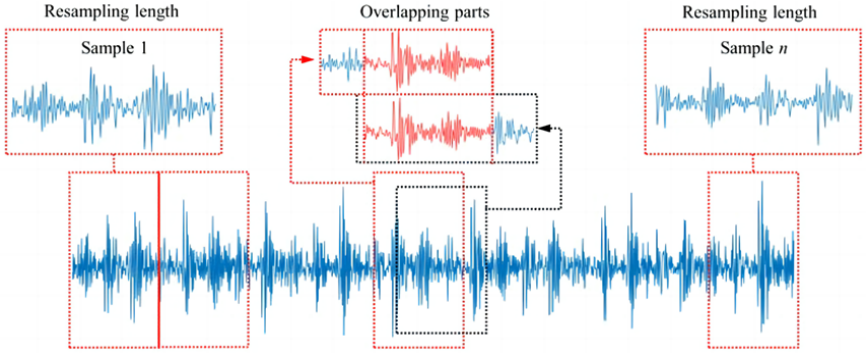

Figure 3. Data set resampling

3.2. XJTU data set

The XJTU dataset is another widely used rolling bearing fault dataset, collected and organized by researchers from the Institute of Design Science and Basic Components at Xi'an Jiaotong University. Unlike the CWRU dataset, the XJTU dataset consists of the entire process of 15 bearings from normal operation to fatigue failure, making it a rolling bearing fatigue fault dataset. The experimental setup for the XJTU dataset includes an alternating current motor, motor speed controller, support shaft, and hydraulic loading system[8]. The vibration signals of the tested bearings are measured using accelerometers installed in both horizontal and vertical directions. The XJTU dataset contains fatigue fault data from 15 LDK UER204 model rolling bearings. In this paper, we selected bearings 3_1, 3_2, 3_4, and 2_3 from the dataset for analysis, which cover four fault modes: outer race fault (OR), inner race fault (IR), cage fault, and combined faults (IBCO). Similar to the CWRU dataset, the original vibration signal data in the XJTU dataset has a sampling frequency of 25.6kHz, which cannot be directly used as input for the TST model. Therefore, it also needs to undergo preprocessing using the resampling method depicted in Figure 3. After preprocessing, the XJTU dataset contains 4000 samples, with 2800 samples forming the training set and 1200 samples forming the test set.

3.3. UCOON data set

The UCOON dataset is a gearbox dataset collected and organized by the University of Connecticut from a two-stage gear transmission box. The input shaft of the gearbox is driven by an AC motor, and the speed of the input shaft is monitored using a tachometer. The first-stage input shaft is equipped with a 32-tooth and an 80-tooth gear, while the second-stage output shaft is equipped with a 48-tooth and a 64-tooth gear[9]. The quiver signal data of the entire gearbox is collected using an measuring device fixed on the gearbox housing, with a sampling frequency of 20kHz. Each sample in the dataset has a sampling length of 3600 data points. To fit the input dimension of the TST model, we selected the middle 2048 data points of each sample as the data for analysis in this chapter. The UCOON dataset consists of 936 samples and contains 9 fault modes, namely Normal Condition (NC), Missing Tooth, Crack, Spall, and 5 different degrees of chipped teeth (referred to as Chip5a to Chip1a). For this paper, we randomly selected 655 samples as the training set and the remaining 281 samples as the test set[10].

4. Simulation analysis

4.1. Fault diagnosis results of TST model on a given data set

This article will provide a detailed discussion and analysis of the fault diagnosis results of the proposed TST model, including the basic fault diagnosis outcomes and a relative to unrelated existing fault diagnosis methods[11]. The hardware environment used to run the code includes an AMD Ryzen ThreadRipperPro 3995WX CPU and an NVIDIA RTX 3090 GPU. The deep learning framework and software environment utilized are Python 3.8, Pytorch 1.8.1, and CUDA 10.2.

During the training process, considering that the CWRU, XJTU, and UCOON datasets have different scales, the batch sizes for the three datasets were set to 128, 64, and 32, respectively[12]. To make the training process more generalized, no model selection strategies like Early Stopping were employed, and the TST model was trained for 50 epochs without early termination. Furthermore, to mitigate the consequences of random starting values, the TST model was repetitively trained 100 times on each dataset with the same parameter settings. The results of the loss function and fault diagnosis accuracy are presented in the form of both average values and box plots with statistical information. Figures 4 to 9 illustrate the training process of the TST model on the three datasets, including the variations of the loss function and recognition accuracy on both the training and testing sets. In the early stages of training, the impact of random initialization could be significant, which is why the changes in the loss function and accuracy are displayed using box plots[12]. However, the overall variations of the loss function and accuracy throughout the entire training process are represented using the average values of the repeated training results.

From the results in Figure 4(a) and Figure 4(b) on the CWRU dataset, it can be observed that different random initializations of parameters lead to significant fluctuations in the model's loss function during the early stages of training. However, after approximately 10 epochs of training, the box plots of the loss function values on both the training and testing sets become more flattened, and the number of outliers (depicted as black dots in Figure 4(a) and Figure 4(b)) noticeably decreases. This indicates that the impact of random initialization has been largely mitigated, and the loss function on the training and testing sets has gradually stabilized. These results demonstrate that the TST model exhibits good robustness in fault diagnosis performance, as it can effectively mitigate the effects of random initialization to a certain extent. In addition, the variation of the average loss function value (denoted as AvgLoss) throughout the entire training process is shown in Figure 4(c). It can be observed that the TST model demonstrates a decaying trend of the loss function values on both the training and testing sets, and there are no significant wide fluctuations. This suggests that the choice of optimizer parameters and the design of the model architecture are reasonably appropriate, resulting in a reliable gradient optimization during the training process. When training concludes, the average loss function values on the training and testing sets are relatively close, indicating that the model did not suffer from severe overfitting. Overall, these findings demonstrate that the TST model exhibits a good level of robustness and generalization in fault diagnosis performance, effectively dealing with the challenges posed by random initialization and achieving a balanced performance between the training and testing sets.

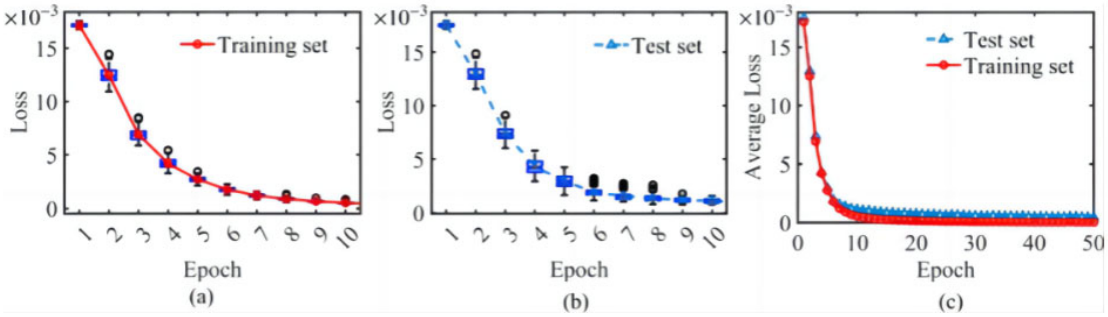

Figure 4. Changes in the loss function on the CWRU data set during the training process (a) Box diagram of the first 10 generations of the loss function in the training set (b) Box diagram of the first 10 generations of the loss function in the test set (c) the average loss function value during the entire training process

Similarly, in Figure 5 and Figure 6, similar phenomena can be observed. However, it should be noted that on the UCOON dataset, the TST model exhibits slightly different variations in the loss function during the training process compared to the other two datasets. From Figure 6(c), it can be observed that the TST model's loss function on the UCOON dataset shows evident periodic decay. Specifically, in the early stages of training, the loss function remains relatively unchanged, stabilizing around a higher value (0.14) until approximately 8 steps of training when a more pronounced decay in the loss function becomes evident. These results suggest that on the UCOON dataset, the TST model's loss function may encounter a larger "plateau" region near the initial parameters, which corresponds to an area with small gradients. The existence of this region slows down the convergence speed of the model, as shown in Figure 6(a) and Figure 6(b). Unlike the CWRU and XJTU datasets, the TST model requires about 20 steps of training on the UCOON dataset to gradually abolish the impact of haphazard initialization. Overall, at the end of training, the TST model achieves average loss function values of 2.95 × 10−5 and 4.55 × 10−4 on the training set and test set of the CWRU dataset, distinctly. On the XJTU dataset, these values are 5.16 × 10−6 and 5.38 × 10−4, respectively. On the UCOON dataset, the values are 3.17 × 10−4 and 1.51 × 10−3, respectively. Overall, the TST model converges to relatively small ranges of loss function values on all three datasets, indicating that the TST model exhibits good convergence characteristics.

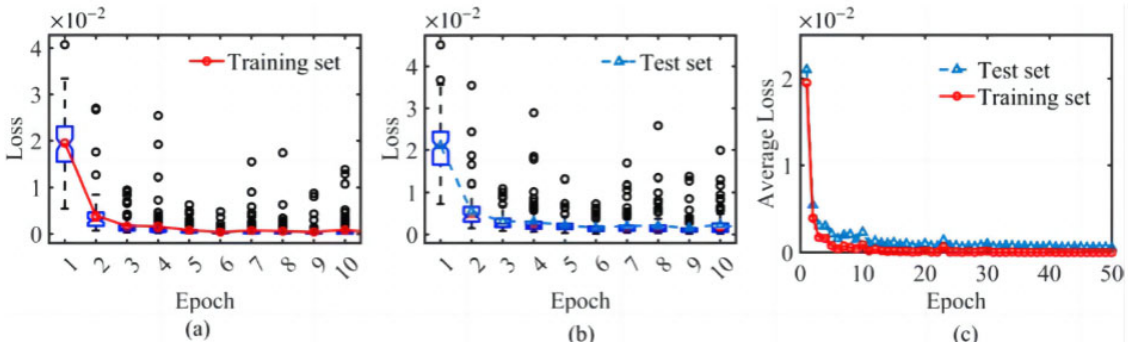

Figure 5. Changes in the loss function on the XJTU data set during the training process (a) Box diagram of the first 10 generations of the loss function in the training set (b) Box diagram of the first 10 generations of the loss function in the test set (c) the average loss function value during the entire training process

Furthermore, Figures 7 to 9 illustrate the variation of fault diagnosis accuracy of the TST (Time Series Transformer) model on three datasets during the training process. Similar to the changes in the loss function, the results are presented in two forms: boxplots representing early training stages and the average values throughout the entire training process. On the CWRU dataset, as shown in Figures 7(a) and 7(b), the accuracy of the TST model on the training and test sets fluctuates significantly during the early stages of training, likely due to the impact of random initialization. However, after approximately 10 training steps, the shape of the boxplots gradually becomes flat, indicating that the influence of random initialization has been mostly eliminated, and the recognition accuracy stabilizes as the training progresses. Moreover, as depicted in Figure 7(c), it can be observed that the average accuracy of the TST model on both the training and test sets increases steadily throughout the entire training process. At the end of training, the TST model achieves high recognition accuracy on both the training and test sets, demonstrating that the approach proposed in this chapter can achieve accurate fault diagnosis.

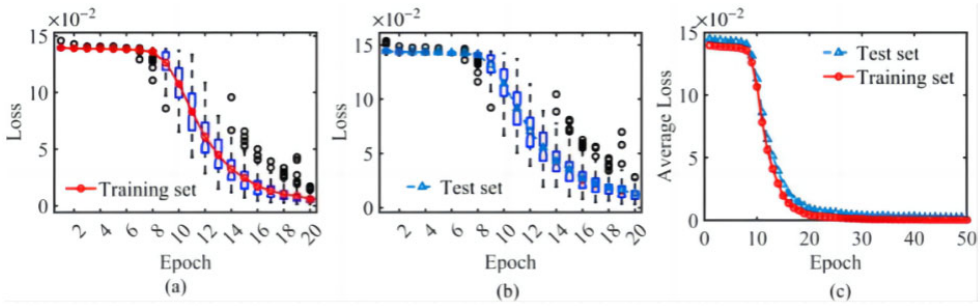

Figure 6. Loss function changes on UCOON data set during the training process (a) Box diagram of the first 20 generations of loss functions in the training set (b) Box diagram of the first 20 generations of loss functions in the test set (c) average loss function values during the entire training process

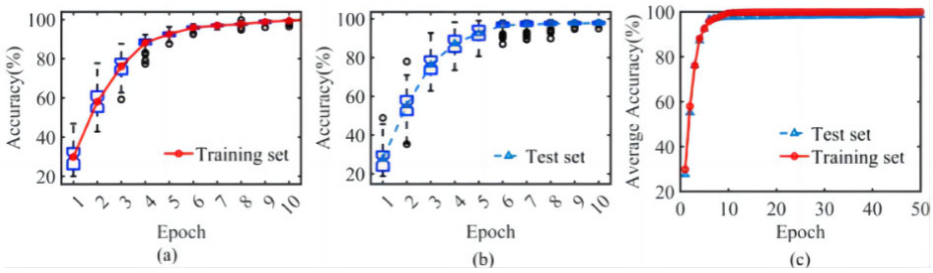

Figure 7. Accuracy changes on the CWRU data set during the training process (a) Box diagram of the first 10 generations of accuracy in the training set (b) Box diagram of the first 10 generations of accuracy in the test set (c) Mean achievement throughout training

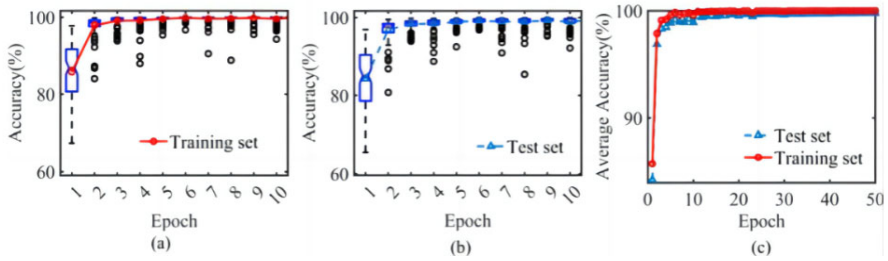

Figure 8. Accuracy changes on the XJTU data set during training (a) Box diagram of the first 10 generations of accuracy in the training set (b) Box diagram of the first 10 generations of accuracy in the test set (c) Mean success rate during training iterations

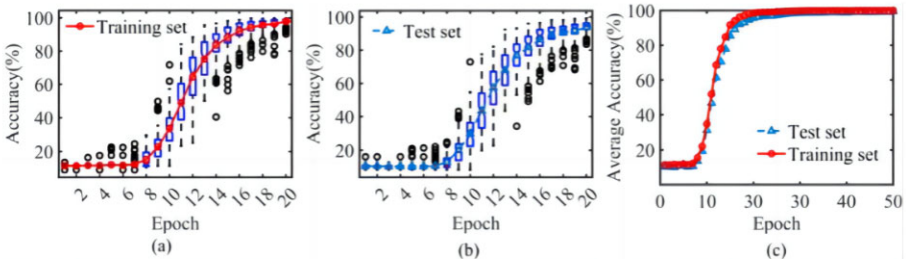

Figure 9. Accuracy changes on the UCOON data set during the training process (a) Box diagram of the first 20 generations of accuracy in the training set (b) Box diagram of the first 20 generations of accuracy in the test set (c) average accuracy during the entire training process

Similar trends are also observed in the results presented in Figures 8 and 9. Additionally, it is essential to note that on the UCOON dataset, homogeneous to the changes within the loss function, the TST model's accuracy exhibits a distinct flat phase during the early stages of training. The recognition accuracy on both the training and test sets remains around 11%, which is approximately equivalent to random guessing. This indicates that the model did not undergo effective parameter updates during the initial training stage, further confirming the existence of a "plateau region" around the initial values. Overall, when training concludes, the TST model achieves average accuracy of 100% on the training set and 98.63% on the test set for the CWRU dataset, 100% on the training set and 99.78% on the test set for the XJTU dataset, and 100% on the training set and 99.51% on the test set for the UCOON dataset. Consequently, the TST model attains an average fault diagnosis accuracy of over 95% across all three datasets, demonstrating its excellent fault diagnosis performance.

4.2. Comparison with other fault diagnosis methods

It can be seen from the results in the table that the TST model proposed in this paper can achieve a higher accuracy than many existing fault diagnosis methods without any

additional preprocessing, which further proves the effectiveness and superiority of the proposed method.

Table 1. Results Of Comparison Between TST Model And Other Existing Fault Diagnosis methods

Model name	Signal processing methods and others	Accuracy rate
TST (Proposed)	Raw vibration signal data	99.72%
NKH-KELM	Multiscale dispersion entropy	95.56%
CWT-CNN	Time-frequency graph obtained by wavelet transform	99.40%
DCN	Raw vibration signal data	99.31%
LSTM	Short-time Fourier transform	98.65%

5.Conclusion

This paper proposes an innovative fault diagnosis model called Time Series Transformer (TST) and demonstrates its effectiveness in the field of fault diagnosis through validation on three experimental datasets: CWRU, XJTU, and UCOON. Compared to existing fault diagnosis methods, the TST model shows significant advantages. Overall, this paper makes a valuable contribution to the field of fault diagnosis by introducing the Time Series Transformer model and effectively addressing various fault diagnosis issues. Future work can build upon this foundation to further refine and expand the model, enabling it to better handle practical engineering fault diagnosis challenges. Additionally, it is essential to consider the model's interpretability and computational efficiency to enhance its applicability and acceptance in real-world scenarios.

References

[1] X. Zhang, Y. Liang, and J. Zhou, et al, "A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM," vol. 69, no. 30, pp. 164-179, 2021.

[2] LEVENT E, TURKER I, and SERKAN K, "A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier," Journal of Signal Processing Systems, vol. 91, no. 20, pp. 179-189, 2019.

[3] H. Fang, J. Deng, and Y. Bai, et al, "Clformer: A lightweight transformer based on convolutional embedding and linear self-attention with strong robustness for fault diagnosis under limited sample conditions," IEEE Transactions on Instrumentation and Measurement, no. 71, pp. 1-8, 2022.

[4] S. Zhou, L. Wu, and H. Su, "Random Exponential attractors for a non-autonomous FitzHugh-Nagumo lattice system with multiplicable white noise," Journal of Zhejiang Normal University, vol. 42, no. 1, pp. 1-8, 2019.

[5] H. Zhang, X. Wang, and X. Li, "Low signal-to-noise ratio noise reduction algorithm based on adaptive threshold active speech detection and minimum mean square error log-spectral amplitude estimation," Journal of Research in Science and Engineering, vol. 40, no. 6, pp. 1763-1768, 2019.

[6] X. Ma, J. Yu, and F. Yang, et al, "Pseudo-random sequence generator based on high dimensional chaotic system," Journal of Dalian Institute of Light Industry, vol. 39, no. 2, pp. 143-149, 2020.

[7] Z. Jiang, and Q. Yang, "Generalizedcodeindexmod-ulationbasedondirectsequencespreadspectrum," Application Research of Computers, vol. 36, no. 4, pp. 1186-1188, 2019.

[8] Jafarinejad F, Pouyan A A, "A modular synthesis approach for intelligent manufacturing system design: A Petri net based transformation method," Signal Processing and Intelligent Systems Conference (SPIS), vol. 33, no. 5, pp. 40-45, 2016.

[9] KUEHN I, CORDIER J J, and BAYLARD C, et al, "Management of the ITER buildings configuration for the construction and installation phase," IEEE Transactions on Plasma Science, vol. 46, no. 1, pp. 194-200, 2018.

[10] Somes N, and Moore J, "Design and Construction of a Regional Scale Bioret ention System," Rainwater&Urban Design, vol. 20, no. 3, pp. 12-13, 2007.

[11] Urban K L , Scheller F , and Bruckner T, "Suitability assessment of models in the industrial energy system design," Renewable and Sustainable Energy Reviews, vol. 137, pp. 110400-110402, 2021.

[12] Kamrowska-Zauska D , "Impact of AI-Based Tools and Urban Big Data Analytics on the Design and Planning of Cities," Land, vol. 10, pp. 101-103, 2021.

[13] L. Deng, "Analysis on the application and strategy of sponge city concept in landscape architecture planning," Smart Cities, vol. 6, no. 5, pp. 143-144, 2020.